Appearances of the Birthday Paradox in High Performance Computing

Victor Eijkhout, Margaret Myers, John McCalpin
Texas Advanced Computing Center, Department of Computer Science; UT Austin

TACC

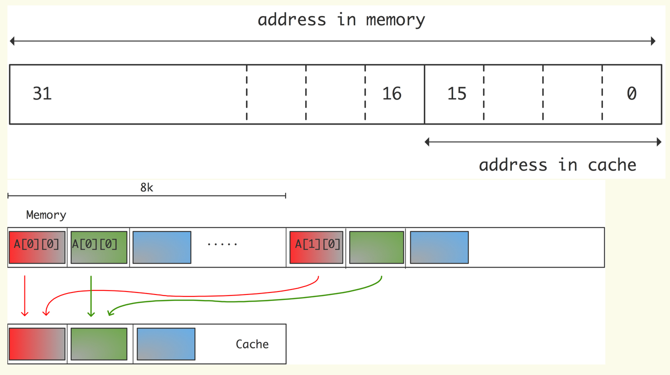
Of pigeons and performance

- Assignment of a few pigeons to more holes: it should fit
- Random assignment: even with very few pigeons you get collisions
- HPC angle: collisions are bad for performance.

Let's consider two examples, and analyze them with elementary statistics.

Cache mapping

- Cache is small fast memory for re-used data
- Mapping problem: memory addresses to cache addresses
- Birthday problem: what is the chance that two random addresses map to the same cache address



Research question

- What is the effective cache capacity; given cache size $N = m \cdot k$ (m #sets, k associativity), after mapping N random addresses how many remain in the cache?
- What is a safe effective cache size: for what $N' < N$ is the probability of conflict $< p$?

Expected working set size

Cache size 1k, number of elements stored:

associativity	1	2	3	4	10	50	100
expected working set size	632	729	775	805	875	945	962

Cache size 4k, 4-way associative, probability of no conflict:

cache size	100	200	500	1000	2000
no-conflict probability	.222	$4.15 \cdot 10^{-2}$	$2.73 \cdot 10^{-4}$	$6.30 \cdot 10^{-8}$	$3.34 \cdot 10^{-15}$

Intel Xeon processor

64 sets $\times$ 8-way associativity $\times$ 64 bytes per cacheline $= 2^{15}$ bytes.

We expect 441 out of 512 mapped cachelines to remain in the cache.

Derivation, for the interested reader

$$E[\#\text{stored}] = mE[Y_i], \qquad i \in [0, m).$$

Our basic random variables are

$$\begin{cases} X_i & \text{the number of addresses mapped to set } i \\ Y_i & \text{the number of addresses stored in set } i \end{cases}$$

where we note that $Y_i = \min(X_i, k)$.

We observe that Y_i has a maximum value of k, the associativity, but X_i can be larger.

$$\begin{aligned} E[Y_i] &= \sum_{j=0}^{k-1} jP(Y_i = j) + kP(Y_i = k) \qquad (1) \\ &= \ldots \\ &= k - \sum_{j=0}^{k-1} (k-j) \binom{n}{j} \left(\frac{k}{n}\right)^j \left(\frac{n-k}{n}\right)^{n-j} \end{aligned}$$

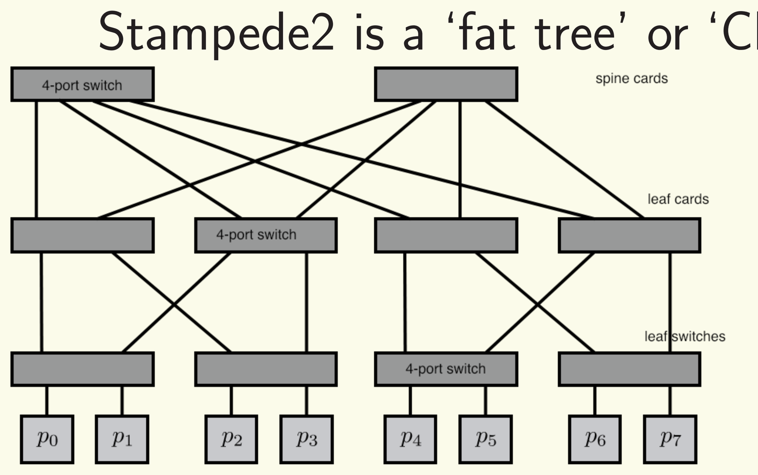
The 'birthday paradox'

It's not a paradox:
it's the statement that collisions of unlikely events are much more likely than you'd think.

What is the chance that two people have the same birthday?
for a 50% chance it is enough to have $n = 23$,
and $n = 70$ gives a 99.9% chance.

Network switches

Stampede2 is a 'fat tree' or 'Clos network'.

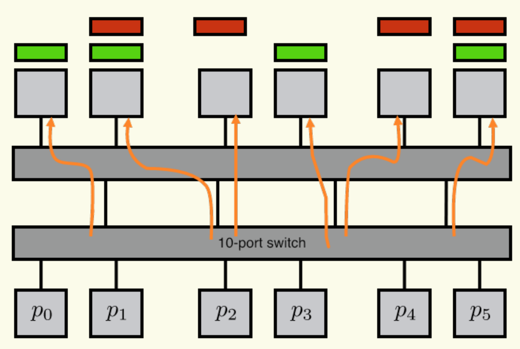


Research question; two complications

1. Output routing: the port is determined statically by the destination.
2. Oversubscription: more inbound wires (k) to the port than outbound (n).

Given $k < n$ messages, what is the chance of conflict?

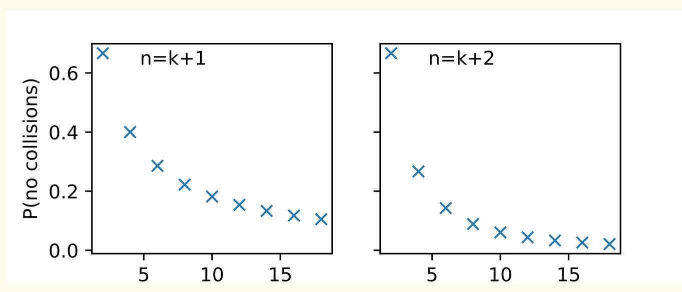
Illustration



Examples of destination processors that are reached without contention (green) and with (red).

Result and analysis

Probability of no collisions is $2^{n-k}/C(n, k)$.



The probability of no message collisions for:
one-way ($n = k + 1$, left) and two-way ($n = k + 2$, right) oversubscription as function of the number of destinations k.

$$\frac{P\big(\text{No collisions for } n\big)}{P\big(\text{No collisions for } n-1\big)} = \frac{2(n-k)}{n}.$$

With $P = 1$ for $n = k$, we get (for $k \geq 2$):

$$\text{if } n = k+1\text{: } P\big(\text{no collisions}\big) = \frac{2}{k+1}$$
$$\text{if } n = k+2\text{: } P\big(\text{no collisions}\big) = \frac{8}{(k+1)(k+2)}$$

With total number of ports $p = n + k$:

$$n = k+2\text{: } P\big(\text{no collisions}\big) = \frac{8}{n(n-1)} = \frac{32}{p(p+2)}.$$

Example: 30-ports, minimal oversubscription is 16 input ports and 14 output ports.

With 14 sources and destinations this gives a chance of no collisions of 1/30.

Publication

https://arxiv.org/abs/1909.12195v1